

Intel[®] Thread Checker 2.2

Release Notes

Contents

[Overview](#)

[Product Contents](#)

[What's New](#)

[System Requirements](#)

[Known Issues and Limitations](#)

[Technical Support](#)

[Related Products](#)

Overview

The Intel(R) Thread Checker can help developers create threaded applications quickly by detecting thread safety issues in programs that use Win32*, POSIX* and OpenMP* threads. Source locations where shared memory or synchronization conflicts could cause consistency or deadlock conditions are shown, including call stacks for each thread.

Product Contents

Intel(R) Thread Checker 2.2 plug-in to the VTune™ Performance Analyzer to be installed on Microsoft Windows* systems.

Intel(R) Thread Checker 2.2 Linux* Remote Agent to be installed on Linux* systems.

What's New

Intel(R) EM64T Support

Source and binary instrumentation of 32-bit applications on processors supporting Intel(R) EM64T running 64-bit operating systems.

Command Line Support

Start a predefined activity from the command line. For more information, see "VTL" in the on-line help.

Diagnostic Suppression

Filter benign or less critical diagnostics to suppress them from appearing in the diagnostic list. Save related filtering rules to a file (.flt) to apply to future activities.

Side-by-Side Source Views

Simultaneously view where threads participate in a data race or deadlock.

Memory Usage Control

User settable limit of the amount of memory used by the Intel(R) Thread Checker analysis.

System Requirements

Microsoft Windows* Systems

Minimum Hardware Required

- Pentium(R) III processor
- 512 MB of RAM
- 300 MB of disk space

Recommended Hardware

- Pentium(R) 4 processor supporting Hyper-Threading Technology or Intel(R) Xeon™ processor or higher
- 1 GB of RAM

Required Software

- Microsoft Windows* 2000, Microsoft Windows* XP Professional or Microsoft Windows* Server 2003
- VTune™ Performance Analyzer 7.2 or higher
- Microsoft Internet Explorer* 6.0 or higher
- Microsoft Visual C++* 6.0 or higher
- Adobe® Reader®*

Required Software for OpenMP* Analysis or Source Instrumentation

- Intel(R) C++ Compiler for Windows* 8.1, Package ID: w_cc_pc_8.1.023 or higher
- Intel(R) Fortran Compiler for Windows* 8.1, Package ID: w_fc_pc_8.1.023 or higher

Pentium(R)-based Linux* Systems

Minimum Hardware Required

- Pentium(R) III processor
- 512 MB of RAM
- 100 MB of disk space

Recommended Hardware

- Pentium(R) 4 processor supporting Hyper-Threading Technology or Intel(R) Xeon™ processor or higher
- 1 GB of RAM

Required Software

- Red Hat* Enterprise Linux* 3.0, Red Hat* Enterprise Linux* 4.0, Red Hat* Application Server 2.1, Red Flag* 4.1, or SUSE LINUX Enterprise Server* 9.0

Required Software for OpenMP* Analysis or Source Instrumentation

- Intel(R) C++ Compiler for Linux* 8.1, Package ID: l_cc_p_8.1.026 or higher
- Intel(R) Fortran Compiler for Linux* 8.1, Package ID: l_fc_p_8.1.023 or higher

Itanium(R)-based Linux* Systems

Minimum Hardware Required

- Intel(R) Itanium(R) processor
- 512 MB of RAM
- 100 MB of disk space

Recommended Hardware

- Intel(R) Itanium(R) 2 processor or higher
- 1 GB of RAM

Required Software

- Red Hat* Enterprise Linux* 3.0, Red Hat* Enterprise Linux* 4.0, Red Flag* 4.1, or SUSE LINUX Enterprise Server* 9.0
- Intel(R) C++ Compiler for Linux* 9.0, Package ID: l_cc_p_9.0.021 or higher
- Intel(R) Fortran Compiler for Linux* 9.0, Package ID: l_fc_p_9.0.021 or higher

Known Issues and Limitations

This product already contains the VTune(TM) Performance Analyzer Patch for Intel(R) Pentium(R) Processor Extreme Edition and Intel(R) Pentium(R) D Processor (vpa_7.2_pp1.zip). Installing any patch, which requires manual copying of files regardless of version number, could potentially overwrite newer versions of these files. Please contact customer support if you wish to install a patch that requires manual file copying.

The binary instrumentation technology used by the Intel(R) Thread Checker can cause some applications to change behavior, or terminate abnormally. In this case, you may be able to complete the analysis after lowering the instrumentation levels of the problematic modules.

The Intel(R) Thread Checker does not detect threading errors for threads and synchronization objects shared across processes (executables).

The Intel(R) Thread Checker cannot analyze processes that are already running. You must either specify the executable as the application to launch, or specify it as a module of interest and then launch it manually when prompted.

This Intel(R) Thread Checker supports analysis only of native binaries and does not support intermediate executable representations intended for managed runtime environments.

Source and binary instrumentation of 64-bit applications on processors supporting Intel(R) EM64T is not supported by this release.

If an application has data races on a global variable, the definition information of the variable will not be available.

The Intel(R) Thread Checker ignores the Duration setting in any of the Activity configuration dialogs.

After uninstalling the Intel(R) Thread Checker, the standard project wizards may not appear in the New Project dialog. In this case, restart the VTune™ environment to restore the standard project wizards.

This release does not support Unicode file names.

Software that defines functions with names that match system API names such as ReadFile() or CreateThread() may have link failures when built with source instrumentation using /Qtcheck (or -tcheck on Linux*) and the software may crash when run with Thread Checker. Software should use function names that do not match system API names to avoid these issues.

When using source instrumentation with functions that contain inline assembly, invalid diagnostics may be generated if data is collected outside of the VTune™ environment. To avoid this, compile without /Qtcheck (or -tcheck on Linux*) and run your software in the VTune™ environment.

Incorrect diagnostics may sometimes be produced when using the Intel(R) C++ compiler 7.x source instrumentation. To avoid this, use binary instrumentation by compiling without /Qtcheck and run your software in the VTune™ environment.

Use of Stop, Pause/Resume or Cancel to terminate instrumentation or data collection before it completes may cause the VTune™ environment to become unresponsive especially when using the Linux* Remote Agent.

The first time an Intel(R) Thread Checker 1.0 project is opened with the Intel(R) Thread Checker 2.0 or higher, you may see a warning dialog: "The following were present when this file was created but are not currently present: Intel(R) Thread Checker. There may be unexpected failures if you proceed." Select "Yes" to proceed and no failures will occur.

Although Get Tuning Advice (F8 or View menu) may appear to be available when viewing Intel(R) Thread Checker results, it should not be used as it has been found to cause the VTune™ environment to crash in certain instances.

When opening help topics that contain Related Topics button links, you may see an Internet Explorer* warning message that reads: "An ActiveX control on this page might be unsafe to interact with other parts of the page. Do you want to allow this interaction?". You can safely click Yes to continue. This problem occurs due to registry errors caused by installing a Windows* SP. To avoid seeing the warning, you can reregister the HTML Help ActiveX control. To do this, execute the following two commands:

```
regsvr32 /u %windir%\system32\hhctrl.ocx  
regsvr32 %windir%\system32\hhctrl.ocx
```

Microsoft Windows* Analysis

Important Note: The Intel(R) Thread Checker must be reinstalled after installing or upgrading an Intel(R) Compiler.

When using the Intel(R) Thread Checker from within the Microsoft Visual Studio* environment, the Cancel and Stop commands will not shut down the process under analysis as expected.

When using the Intel(R) Thread Checker from within the Microsoft Visual Studio* environment, some output messages may not be generated.

Microsoft DirectX DirectShow* SDK: When using versions of DirectShow that are earlier than 9.0, you may need to manually raise the instrumentation level of QUARTZ.DLL from Module Imports to API Imports. This is done via the Advanced Activity Configuration dialog, Thread Checker Configure... button, Module Instrumentation tab, Instrumentation Level cell drop down list.

Linux* Analysis

Important Note: The Intel(R) Thread Checker Linux* Remote Agent must be reinstalled after installing or upgrading an Intel(R) Compiler for Linux*.

This release does not support the following types of Linux* binaries:

- Linux applications statically linked with C/C++ runtime library or POSIX thread library.
- The Linux* kernel and any kernel modules.
- The Linux* dynamic linker/loader (/lib/ld-linux.so.*, /lib/ld-*.so).

Microsoft Windows* XP SP2 enforces network firewall restrictions by default. Remote collection activities will request network access. Please unblock the firewall for the following processes when requested:

- **VTune\Shared\Bin\VTuneEnv.exe**
- **Vtune\tcheck\bistro_cgc\ATLBistroController_RLinuxKSL32.exe**
- **Vtune\tcheck\bistro_cgc\CGExeCtrl_RLinuxKSL32.exe**

If there is a firewall between the host Microsoft Windows* machine and the remote Linux* machine, the data collection may fail.

If you open the results from a previously collected remote activity without an active Linux* Remote Agent (ittserver process), you will see a few related information messages. If you click OK on each of them, you will then be able to work with the results as usual.

If you delete the %VTUNE_GLOBAL_DIR%\VTMH folder (typically "C:\Program Files\Intel\VTune\global_data\VTMH"), previously collected remote activity results will not show source code locations due to missing symbolic information. To preserve the symbolic information associated with a remote activity, create a Pack and Go file (.vxp) of the activity before deleting the %

VTUNE_GLOBAL_DIR%\VTMH folder.

If the ittserver process is killed (Ctrl-C, etc.) during a data collection, the VTune™ environment may become unresponsive.

To use source instrumentation on Linux*, please ensure that you setup your environment variables by running the Intel(R) Thread Checker environment variable script, for example,

```
source /opt/intel/itt/tcheck/bin/32/tcvars.sh or  
source /opt/intel/itt/tcheck/bin/64/tcvars.sh before starting the ittserver.
```

If the data directory (specified using: `/opt/intel/itt/bin/32/ittserver [-d dataDirectory]`) matches the cache directory (specified in the VTune™ environment), the ittserver or the Linux* operating system may crash or no results may be generated. In addition, the data directory should be unique for each Linux* account under which ittserver is launched.

Note: If a data directory is not specified, it defaults to the current working directory.

You should recreate an activity rather than use another run of an existing activity in the following two cases:

1. After changing the set of linked libraries, either explicitly or implicitly, for example, changing the LD_LIBRARY_PATH environment variable.
2. After replacing a executable module of the same name with an older modification time.

After changing the LD_ASSUME_KERNEL environment variable, you should clear the Linux* Remote Agent instrumentation cache and recreate the activity.

If you recompile a binary with debug information, it's instrumentation level will not be increased automatically. In this case, you should recreate the activity or change it manually. This is done using the Advanced Activity Configuration dialog, Thread Checker Configure... button, Module Instrumentation tab, Instrumentation Level cell drop down list.

Asynchronous signal handling is not fully supported in this release. Programs that use asynchronous signals may stall when analyzed with the Intel(R) Thread Checker.

When an instrumented application calls the **exec** system call, the whole image context is replaced with the new image. If the application calls to **exec** with the name of the instrumented image, then the instrumented image will be used. In the other cases, the original (non-instrumented) image is called, and no results will be generated from this point in the run.

The Intel(R) Thread Checker Linux* Remote Agent does not support setuid images. The setuid mechanism is used to give a user process the effective user ID of another user, usually root. You can use the Intel(R) Thread Checker Linux* Remote Agent and run the setuid executable only if you are logged in as the same user as the owner of the setuid executable.

The application cannot be a command-line application that expects interactive user input.

If pthread_exit() is called from the main thread, and if actual deadlock and stall detection or real time diagnostic display is turned on and forceful program termination upon deadlock or program exit is turned off, the program under Thread Checker analysis may hang. If this happens, please turn off

actual deadlock and stall detection and real time diagnostic display or turn on forceful program termination upon deadlock or program exit.

Source and Binary instrumentation may dramatically increase the usage of stack space. On Linux* you use the "limit" command or "ulimit" if you are using a different shell (bash) to specify the stack space used for applications. This limit may need to be increased to get programs to run. Be careful with unlimited stack space specification. This sometimes does not leave any space to create the pthread stacks, which can cause them to fail.

When running on an Itanium(R) processor, binary instrumentation on Linux* is not supported with this release. Therefore, you should not change the Module instrumentation options in the collector configuration dialog.

Technical Support

The product support web site (<http://support.intel.com/support/performancetools/threadchecker/>) contains frequently asked questions, product documentation, product errata, as well as solutions to many known issues.

To receive technical support for this product and product updates, you need to register for an Intel(R) Premier Support account at the Intel(R) Registration Center (<http://www.intel.com/software/products/registrationcenter/>).

When submitting issues to Intel(R) Premier Support (<https://premier.intel.com/>), enter a product name of *Intel Thread Checker* and provide the product build number found in the ThreadCheckerSupport.txt file. This is done using Start, Programs, Intel(R) Software Development Tools, Intel(R) Thread Checker, View Support and Build Ids. If your issue involves the Intel(R) Thread Checker Linux* Remote Agent, please include the corresponding build number from the ThreadCheckerSupport.txt file on the Linux* system, which is in the directory where the Intel(R) Thread Checker Linux* Remote Agent was installed. The default installation directory is "/opt/intel/itt".

Once you have contacted us with your suggestion or problem using your Premier Support account, a technical support engineer will respond within one (1) Intel business day.

If you have not received, lost your login ID or password, or are having trouble with access, please visit <https://registrationcenter.intel.com/support>.

Related Products and Services

Information on Intel(R) Software Development Products is available at <http://www.intel.com/software/products>.

Some of the related products include:

- The Intel(R) Thread Profiler is a performance tuning tool for parallel programs that use Win32*, POSIX*, OpenMP* or custom synchronization.

- The VTune™ Performance Analyzer enables you to evaluate how your application is utilizing the CPU and helps you determine if there are modifications you can make to improve your application's performance.
- The Intel(R) Compilers are an important part of making software run at top speeds with full support for the latest Pentium(R) and Itanium(R) processors.
- The Intel(R) Cluster Tools can help developers create, analyze and optimize high-performance applications on clusters of Intel(R) processor-based systems.
- The Intel(R) Performance Library Suite provides a set of routines optimized for various Intel processors.
- The Intel(R) Software College provides training for developers on leading-edge software development technologies. Training consists of online and instructor-led courses covering all Intel architectures, platforms, tools, and technologies.

Disclaimer and Legal Information

The information in this document is subject to change without notice and Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document. This document and the software described in it are furnished under license and may only be used or copied in accordance with the terms of the license. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. The information in this document is provided in connection with Intel products and should not be construed as a commitment by Intel Corporation.

EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The software described in this document may contain software defects which may cause the product to deviate from published specifications. Current characterized software defects are available on request.

Intel, the Intel logo, Intel SpeedStep, Intel NetBurst, Intel NetStructure, MMX, i386, i486, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Celeron, Intel Centrino, Intel Xeon, Intel XScale, Itanium, Pentium, Pentium II Xeon, Pentium III Xeon, Pentium M, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others.

Copyright (c) Intel Corporation 2002-2005.